

Solving Two-Stage Stochastic Steiner Tree Problems by Two-Stage Branch-and-Cut^{*}

Immanuel Bomze¹, Markus Chimani², Michael Jünger^{3*},
Ivana Ljubić^{1**}, Petra Mutzel^{4*}, and Bernd Zey⁴

¹ Faculty of Business, Economics and Statistics, University of Vienna
{immanuel.bomze, ivana.ljubic}@univie.ac.at

² Institute of Computer Science, Friedrich-Schiller-University of Jena
markus.chimani@uni-jena.de

³ Department of Computer Science, University of Cologne
mjuenger@informatik.uni-koeln.de

⁴ Department of Computer Science, TU Dortmund
{petra.mutzel, bernd.zey}@tu-dortmund.de

Abstract. We consider the Steiner tree problem under a two-stage stochastic model with recourse and finitely many scenarios. In this problem, edges are purchased in the first stage when only probabilistic information on the set of terminals and the future edge costs is known. In the second stage, one of the given scenarios is realized and additional edges are purchased in order to interconnect the set of (now known) terminals. The goal is to decide on the set of edges to be purchased in the first stage while minimizing the overall expected cost of the solution. We provide a new semi-directed cut-set based integer programming formulation, which is stronger than the previously known undirected model. We suggest a two-stage branch-and-cut (B&C) approach in which L-shaped and integer-L-shaped cuts are generated. In our computational study we compare the performance of two variants of our algorithm with that of a B&C algorithm for the extensive form of the deterministic equivalent (*EF*). We show that, as the number of scenarios increases, the new approach significantly outperforms the (*EF*) approach.

Keywords: stochastic Steiner tree, stochastic integer programming, branch-and-cut, Benders decomposition

1 Introduction

Motivation. The classical Steiner tree problem in graphs is a quite well-studied combinatorial optimization problem and has a wide range of

^{*} M.J. and P.M. gratefully acknowledge the hospitality they enjoyed during their stay as visiting research professors at the University of Vienna; much of this research was done during this stay

^{**} Supported by the Hertha-Firnberg Fellowship of the Austrian Science Foundation (FWF)

applications: from the planning of various infrastructure networks (e.g., communication or energy supply) to the study of protein-interaction networks in bioinformatics. Given an undirected graph $G = (V, E)$ with edge weights (*costs*) $c_e \geq 0$, for all $e \in E$, and a subset of required vertices (*terminals*) $R \subseteq V$, the problem consists of finding a subset of edges that interconnects all the terminals at minimum (edge installation) cost.

In practice, however, network planners are often faced with uncertainty with respect to the input data. The actual demand patterns become known only after the network has been built. In that case, networks found by solving an instance in which it is assumed that the complete knowledge of the input is known up-front, might not provide appropriate solutions if deviations from the assumed scenario are encountered. Stochastic optimization is a promising way to take uncertainties into account.

Our problem. We consider the two-stage stochastic Steiner tree problem (SSTP) with fixed recourse and finitely many scenarios in which the terminal set and the edge installation costs are subject to uncertainty. In this model, some edges are purchased in the first stage when only probabilistic information about the second stage is known. In the second stage, one of the given scenarios is realized and additional edges are purchased in order to interconnect the (now known) terminal nodes. The costs of these edges in the second stage are higher than their costs in the first stage. Our goal is to make a decision about edges to be purchased in the first stage, while minimizing the *expected cost* of the solution, which is given as the sum of the purchased edges in the first-stage plus the expected costs of purchasing additional edges in the second stage.

Previous work. Gupta et al. started a series of papers on approximation algorithms for the SSTP. E.g., they have provided a constant factor approximation algorithm for the SSTP in the case that the second stage costs are determined from the first stage costs by multiplication with a fixed inflation factor [5]. The algorithm is based on a primal-dual algorithm that is guided by a relaxed integer linear programming (ILP) solution. For the general case that we consider, Gupta et al. [4] have shown that the problem becomes as hard as Label Cover (which is $\Omega(2^{\log^{1-\epsilon} n})$ -hard). We are not aware of any computational study concerning the SSTP.

Our Contribution. The ILP model used in [5] is based on an undirected cut-set formulation for Steiner trees. We suggest a new semi-directed ILP model and show that it is stronger than the undirected one. We show that the recourse function decomposes into a set of independent *restricted*

Steiner arborescence problems. To solve the problem, we use a Benders-like decomposition method. We nest a branch-and-cut framework for solving the subproblems [8] into a branch-and-cut master approach in which L-shaped and integer-L-shaped cuts are generated [2,7]. In our computational experiments, we study the behaviour of our two-stage branch-and-cut (B&C) algorithm for two differently decomposed variants of our semi-directed ILP model and compare it to solving the deterministic equivalent (EF) directly. It is the first time that the stochastic Steiner tree problem is studied computationally. We report optimal results for SSTP instances with up to 165 vertices and 274 edges.

2 ILP models

2.1 Problem definition

We consider the following two-stage stochastic Steiner tree problem. Let $G = (V, E)$ be an undirected network with a selected root r and with known first-stage edge costs $c_e \geq 0$, for all $e \in E$; let $V_r := V \setminus \{r\}$. The set of terminals, as well as the costs of edges to be purchased in the second stage, is known only in the second stage. These values together form a random variable ξ , for which we assume that it has a finite support. It can therefore be modeled using a finite set of scenarios $\mathcal{K} = \{1, \dots, K\}$, $K \geq 1$. The realization probability of each scenario is given by $p_k > 0$, $k \in \mathcal{K}$; we have $\sum_{k \in \mathcal{K}} p_k = 1$. Denote by $q_e^k \geq 0$ the cost of an edge $e \in E$ if it is bought in the second stage, under scenario $k \in \mathcal{K}$. Denote the *expected second stage cost* of an edge $e \in E$ by $q_e^* := \sum_{k \in \mathcal{K}} p_k q_e^k$. We assume that $q_e^* > c_e$, for all $e \in E$. Furthermore, let $R_k \subseteq V_r$ be the set of terminals under the k -th scenario. We denote by E_0 the set of edges purchased in the first-stage, and by E_k the set of additional edges purchased under scenario k , $k \in \mathcal{K}$.

The *SSTP problem* can then be formulated as follows: Determine the subset of edges $E_0 \subseteq E$ to be purchased in the first stage, so that the overall cost defined as

$$\sum_{e \in E_0} c_e + \sum_{k \in \mathcal{K}} p_k \sum_{e \in E_k} q_e^k,$$

is minimized, while $E_0 \cup E_k$ spans R_k for all $k \in \mathcal{K}$.

Obviously, the optimal first-stage solution of the SSTP is not necessarily a tree [5]. In fact, the optimal solution might contain several disjoint fragments, depending on the subsets of terminals throughout different scenarios, or depending on the second-stage cost structure.

2.2 Undirected model, deterministic equivalent

A deterministic equivalent (in extensive form) of the stochastic Steiner tree problem has been originally proposed in [5]. The authors developed an undirected ILP formulation as a natural extension of the undirected cut-set model for Steiner trees. We briefly recall this model here. The following two sets of binary variables are used in this model:

$$X_e = \begin{cases} 1, & \text{if } e \in E_0 \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad Y_e^k = \begin{cases} 1, & \text{if } e \in E_k \\ 0, & \text{otherwise} \end{cases} \quad \forall e \in E$$

For $D \subseteq E$, let $(X + Y^k)(D) = \sum_{e \in D} (X_e + Y_e^k)$. For $S \subseteq V$, let $\delta(S) = \{\{i, j\} \in E \mid i \in S \text{ and } j \notin S\}$. A *deterministic equivalent* of the SSTP can then be written as:

$$(SSTP_u) \quad \min_{X \in \{0,1\}^{|E|}, Y \in \{0,1\}^{|\mathcal{K}| \times |E|}} \left\{ \sum_{e \in E} c_e X_e + \sum_{k \in \mathcal{K}} p_k \sum_{e \in E} q_e^k Y_e^k \mid \right. \\ \left. (X + Y^k)(\delta(S)) \geq 1, \forall S \subseteq V_r, S \cap R_k \neq \emptyset, \forall k \in \mathcal{K} \right\}$$

Gupta et al. [5] have shown that the LP-solution of the above model can be rounded to a feasible solution with value of at most 40 times that of the optimal solution, if the edge costs in the second stage are given by $q_e^k = \sigma_k c_e$, for all $e \in E$, $k \in \mathcal{K}$, for some fixed scalar σ_k .

2.3 Semi-directed model, deterministic equivalent

It is well known that directed models for Steiner trees provide better lower LP-bounds (see, e.g., [3]), and therefore the natural question arises whether we can extend the model $(SSTP_u)$ by bi-directing the given graph G and replacing edge- by arc-variables in the same model. The main difficulty with the stochastic Steiner tree problem is that the arcs of the first-stage solution cannot be derived using this technique. It is not difficult to imagine an instance in which an edge $\{i, j\} \in E$ is used in direction (i, j) for one scenario, and in the opposite direction (j, i) for another scenario.

Cut-set formulation. Despite the difficulty mentioned above, we can model SSTP using oriented edges to describe the second stage solutions. In other words, we are looking for the optimal first-stage solution (an undirected subgraph of G) such that each solution of scenario k represents a Steiner arborescence rooted at r , whose *arcs* are built upon all the (already installed) first stage *edges* and additional second-stage arcs. In order to

derive the new model, we first bi-direct graph G by defining the set of arcs $A = \{(i, j) \cup (j, i) \mid \{i, j\} \in E, i, j \neq r\} \cup \{(r, i) \mid \{r, i\} \in E\}$. Denote by A_k the arcs of the optimal solution of scenario k , $k \in \mathcal{K}$. For each scenario $k \in \mathcal{K}$, we now introduce binary arc-variables z_{ij}^k , for all $(i, j) \in A$. A variable z_{ij}^k is set to 1 iff the final solution after the second stage in scenario k uses the arc (i, j) . Note that for edges bought in the first stage, each scenario solution has to select one of its corresponding arcs. The new *semi-directed deterministic equivalent* (EF) of the SSTP can then be written as:

$$(EF) \quad \min \sum_{e \in E} c_e X_e + \sum_{k \in \mathcal{K}} p_k \sum_{e=\{i,j\} \in E} q_e^k (z_{ij}^k + z_{ji}^k - X_e)$$

$$\text{s.t.} \quad z^k(\delta^-(S)) \geq 1, \quad \forall S \subseteq V_r, S \cap R_k \neq \emptyset, \forall k \in \mathcal{K} \quad (1)$$

$$z_{ij}^k + z_{ji}^k \geq X_e, \quad \forall e = \{i, j\} \in E, \forall k \in \mathcal{K} \quad (2)$$

$$z_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in A, \forall k \in \mathcal{K} \quad (3)$$

$$0 \leq X_e \leq 1, \quad \forall e \in E, \forall k \in \mathcal{K} \quad (4)$$

Here, $\delta^-(S) = \{(i, j) \in A \mid i \notin S, j \in S\}$. Constraints (1) ensure that for each terminal $v \in R_k$, there is a directed path (using the second stage arcs) from r to v . Inequalities (2) are capacity constraints ensuring that at least one second stage arc is installed for every edge purchased in the first stage. Proofs for the following two lemmata can be found in the Appendix.

Lemma 1. *Formulation (EF) models the deterministic equivalent of the stochastic Steiner tree problem correctly. In particular, in every optimal solution of the model, variables X_e take value 0 or 1.*

Obviously, the semi-directed formulation (EF) for the stochastic Steiner tree problem is at least as strong as the undirected formulation. We can show that the new formulation is even strictly stronger.

Lemma 2. *Denote by $\text{Proj}_{X,Y}(EF)$ the projection of the polytope defined by the LP-relaxation of (EF) onto the space of X and Y variables in which $Y_e^k = z_{ij}^k + z_{ji}^k - X_e$, for all $e = \{i, j\} \in E$, for all $k \in \mathcal{K}$. Let \mathcal{P}_u be the polytope defined by the LP-relaxation of $(SSTP_u)$. Then for any instance of SSTP we have $\text{Proj}_{X,Y}(EF) \subseteq \mathcal{P}_u$ and there are instances for which strict inequality holds and the optimal LP-relaxation value of (EF) is strictly larger than the corresponding LP-relaxation value of $(SSTP_u)$.*

3 Algorithmic Framework

3.1 Decomposition of the (EF) model

The large number of decision variables makes the extensive form (EF) very difficult to solve when considering many scenarios. However, we can rewrite the (EF) formulation as:

$$\min_{X \in \{0,1\}^{|E|}} c^t X + Q(X)$$

in which the so-called *recourse function* $Q(X)$ decomposes into K independent problems, i.e., $Q(X) = \mathbb{E}Q(X, \xi) = \sum_{k \in \mathcal{K}} p_k Q(X, k)$. For a fixed vector \tilde{X} , the k -th subproblem $Q(\tilde{X}, k)$ corresponds to the following NP-hard *restricted Steiner arborescence problem*:

$$(STP_{sd}^k) \quad \min \sum_{e=\{i,j\} \in E} q_e^k (z_{ij}^k + z_{ji}^k - \tilde{X}_e)$$

$$\text{s.t.} \quad z^k(\delta^-(S)) \geq 1, \quad \forall S \subseteq V_r, S \cap R_k \neq \emptyset \quad (5)$$

$$z_{ij}^k + z_{ji}^k \geq \tilde{X}_e, \quad \forall e = \{i, j\} \in E \quad (6)$$

$$z_{ij}^k \in \{0, 1\}, \quad \forall (i, j) \in A \quad (7)$$

Due to the integrality restrictions on the second stage variables, the recourse function $Q(X)$ is non-convex and discontinuous.

3.2 A two-stage Branch-and-Cut approach.

The key idea is to apply a *nested* Branch-and-Cut approach: a Benders-like decomposition method determines the *Master Branch-and-Cut Framework*. Let the following problem be the relaxed master problem (RMP):

$$(RMP) \quad \min_{X \in [0,1]^{|E|}, \Theta_k \geq 0} \{c^t X + \Theta \mid \Theta = \sum_{k \in \mathcal{K}} p_k \Theta_k,$$

a set of L-shaped cuts and integer L-shaped cuts\}.

For a given first stage solution in X , the variables Θ_k are estimated second stage costs of scenario k needed for purchasing *additional* arcs in the second stage in order to interconnect the terminals from R_k . As *optimality cuts* we use *L-shaped* and *integer L-shaped* cuts [2,7] to guarantee the convergence of the algorithm as described below. Observe that no *feasibility cuts* are needed, since we are dealing with the problem with *complete recourse*, i.e., every first-stage solution is feasible.

- Step 0: Initialization.** $UB = +\infty$ (global upper bound, corresponding to a feasible solution), $\nu = 0$. Create the first pendant node. In the initial (RMP) , the set of (integer) L-shaped cuts is empty.
- Step 1: Selection.** Select a pendant node from the B&C tree, if such exists, otherwise STOP.
- Step 2: Separation.** Solve (RMP) at the current node. $\nu = \nu + 1$. Let $(X^\nu, \Theta_1^\nu, \dots, \Theta_K^\nu)$ be the current optimal solution, $\Theta^\nu = \sum_{k \in \mathcal{K}} p_k \Theta_k^\nu$.
- (2.1)** If $c^t X^\nu + \Theta^\nu > UB$ fathom the current node and goto Step 1.
- (2.2) Search for violated L-shaped cuts:**
For all $k \in \mathcal{K}$, compute the LP-relaxation value $R(X^\nu, k)$ of (STP_{sd}^k) .
If $R(X^\nu, k) > \Theta_k^\nu$: insert L-shaped cut (8) into (RMP) .
If at least one L-shaped cut was inserted goto Step 2.
- (2.3) If X is binary, search for violated integer L-shaped cuts:**
(2.3.1) For all $k \in \mathcal{K}$ s.t. z^k is not binary in the previously computed LP-relaxation, solve (STP_{sd}^k) to optimality. Let $Q(X^\nu, k)$ be the optimal (STP_{sd}^k) value. If $\sum_{k \in \mathcal{K}} p_k Q(X^\nu, k) > \Theta^\nu$ insert integer L-shaped cut (9) into (RMP) . Goto Step 2.
(2.3.2) $UB = \min(UB, c^t X^\nu + \Theta^\nu)$. Fathom the current node and goto Step 1.
- Step 3: Branching.** Using a branching criterion, create two nodes, append them to the list of pendant nodes, goto Step 1.

The algorithm described above is a B&C approach in which each of the subproblems (STP_{sd}^k) is solved to optimality using another B&C. This explains the name *two-stage branch-and-cut*.

L-shaped cuts. To solve the LP-relaxation of the (EF) formulation via the models (STP_{sd}^k) given above, we will relax the integrality constraints (7) to $0 \leq z_{ij}^k$, for all $(i, j) \in A$, for all $k \in \mathcal{K}$. Only a small number among the exponential number of cuts will be needed to solve the LP-relaxations (cf. cutting plane method). Therefore, in the corresponding dual problems only those dual variables associated to cuts found in the cutting plane phase will be of interest. We associate dual variables α_S^k to constraints (5) and β_e^k to (6).

Denote by $(\tilde{\alpha}^k, \tilde{\beta}^k)$ the optimal solutions of the dual of the k -th subproblem. Instead of inserting one optimality cut per iteration, we will consider the *multicut* version of the L-shaped method for this problem [1]. This multicut approach applies a disaggregation of optimality cuts per each single scenario. Therefore, the number of master iterations may be significantly reduced, which is of great importance if the number of scenarios is large, or the recourse function $Q(\tilde{X}, k)$ is difficult to solve. For a fixed

first-stage solution $(\tilde{X}, \tilde{\Theta}_1, \dots, \tilde{\Theta}_K)$, we will solve LP-relaxations of all K scenarios, and insert the following L-shaped cuts:

$$\Theta_k + \sum_{e \in E} (q_e^k - \tilde{\beta}_e^k) X_e \geq \sum_{S \subseteq V_r: S \cap R_k \neq \emptyset} \tilde{\alpha}_S^k, \quad (8)$$

for all $k \in \mathcal{K}$ where $\tilde{\Theta}_k < R(\tilde{X}, k)$.

Integer L-shaped cuts. Let X^ν be a binary first stage solution with its corresponding optimal second stage value $Q(X^\nu) = \sum_{k \in \mathcal{K}} p_k Q(X^\nu, k)$. Let $\mathcal{I}^\nu := \{e \in E : X_e^\nu = 1\}$ be the index set of the edge variables chosen in the first stage, and the constant L be a known lower bound of the recourse function. The general integer optimality cut in the L-shaped scheme [7] cuts off the solution (X^ν, Θ^ν) and can be written as:

$$\Theta \geq (Q(X^\nu) - L) \left(\sum_{e \in \mathcal{I}^\nu} X_e - \sum_{e \in E \setminus \mathcal{I}^\nu} X_e - |\mathcal{I}^\nu| + 1 \right) + L. \quad (9)$$

Solving the subproblems. Each of the K subproblems is solved using a *Subproblem Branch-and-Cut Framework* for the restricted Steiner arborescence problem. The subproblems are solved using the algorithm given in [8], augmented with (6). Cuts found during the separation of one subproblem are then stored in a pool where they can be reused by other subproblems (if applicable).

3.3 Reformulation with negative edge costs in the first stage

Alternatively to above, we can consider the following two objective functions when decomposing the problem: $\min \sum_{e \in E} (c_e - q_e^*) X_e + \sum_{k \in \mathcal{K}} p_k \Theta_k$ for the (RMP) formulation. The second stage subproblem is then decomposable into the following subproblems:

$$(STP_{sd*}^k) \quad Q(\tilde{X}, k) = \min \left\{ \sum_{ij \in A} q_{\{i,j\}}^k z_{ij}^k \mid z_{ij}^k \text{ satisfies (5)–(7)} \right\}.$$

In this formulation, variables Θ_k denote the expected costs for interconnecting terminals from R_k plus purchasing all edges from \tilde{X} in the second stage. The difference in using this decomposition, rather than the one described before, is that the edge costs in the first stage become negative and the initial iterations of the master B&C will therefore select many instead of few edges. The generated L-shaped cuts are then written as

$$\Theta_k - \sum_{e \in E} \tilde{\beta}_e^k X_e \geq \sum_{S \subseteq V_r: S \cap R_k \neq \emptyset} \tilde{\alpha}_S^k. \quad (10)$$

We will see that, from the computational point of view, this second approach significantly outperforms the previous one.

4 Computational Results

All experiments were performed on an Intel Core-i7 2.67GHz Quad Core machine with 12 GB RAM, under Ubuntu 9.04. Each run was performed on a single core. We used ABACUS 3.0 as a generic B&C framework; for solving the LP relaxations we used the commercial package IBM CPLEX (version 10.1) via COIN-Osi 0.102.

Depending on the used decompositions (STP_{sd}^k) and (STP_{sd*}^k) , we denote the implementations of the two-stage B&C algorithms by $2BC$ and $2BC^*$, respectively. Thereby, we use following primal heuristic at the root node of the B&C tree (after each iteration, until we obtain the first upper bound): Round the fractional solution X' to a binary solution X'' . If X'' is cycle free, solve all K subproblems to optimality and obtain a valid upper bound $UB = c^t X'' + \sum_{k \in \mathcal{K}} Q(X'')$. For solving (EF) directly, we implemented a branch-and-cut approach analogous to the one given in [8]; we denote the algorithm by EF .

4.1 Benchmark Instances

The benchmark instances used in our study are derived from deterministic inputs taken from the following two sources.

K and P Groups of Instances. These graphs are instances of the prize-collecting Steiner tree problem, originally generated in [6]. Our inputs are graphs obtained by applying several reduction procedures as explained in [8]. The reduced instances contain up to 91 nodes and 237 edges and are available online [9].

lin Instances. These graphs are instances borrowed from the SteinLib [10]. The graphs contain up to 165 nodes and 274 edges with up to 14 terminals. Although for the deterministic Steiner tree problem these instances appear to be solvable by preprocessing or by dual ascent heuristics, the same techniques cannot be applied straight-forwardly to the corresponding SSTP problems.

Converting Deterministic into Stochastic Inputs. Deterministic Steiner tree input graphs $G = (V, E)$ with edge costs c_e , $e \in E$ are transformed into the SSTP instances as follows:

1. We generate K scenarios. To obtain scenario probabilities p_k , we distribute 1000 points (corresponding to the probability of 1‰, each) among these scenarios randomly (ensuring that each scenario has at least probability 1‰).
2. For each scenario k , we construct R_k by independently picking each terminal or Steiner node with probability 0.3 or 0.05, respectively.
3. Each second stage edge costs q_e^k is randomly (independent, uniform) drawn from $[1.1c_e, 1.3c_e]$.

4.2 Comparing the Deterministic Equivalent vs. Two-Stage Branch-and-Cut Approaches

For the K and P instance groups, we focus on comparing the time to obtain provably optimal solutions, required by our two decomposition-based algorithms $2BC$, $2BC^*$ and the standard approach EF. Figure 1 shows the running times in seconds, averaged over all instances of the corresponding group. We observe that decomposing the problem is not worthwhile for instances with less than 20 scenarios. However, as the number of scenarios increases, the benefit of decomposing is obvious: already with 100 scenarios, EF needs 10 times the running time of the two-stage B&C approaches. In additional experiments with 500 scenarios, EF is not able to solve 6 out of 11 instances within two hours, whereas the two-stage approach $2BC^*$ needs only 510 seconds on average.

We also observe that $2BC^*$ always outperforms $2BC$. In particular for the group K instances with 100–500 scenarios, it is 1.8 times faster. This is because the L-shaped cuts generated by $2BC^*$ are sparser ($\tilde{\beta}_e^k$ are often 0) and numerically more stable than the corresponding cuts generated by $2BC$ (cf. Section 3.1).

Table 1 shows the comparison between EF and the two-stage approach $2BC^*$. Instances `lin01`–`lin06` were used to generate inputs with $K \in \{5, 10, 20, 50\}$ scenarios. Column $|R_{avg}|$ gives the average number of terminals in each scenario; OPT* gives the optimal values (or the best upper bound, if the time limit of 2 hours is reached). We compare the running time in seconds ($t[s]$), the number of branch-and-bound nodes (b&b), the final gap obtained after the time limit of two hours, as well as the overall number of iterations in the B&C (#iter). We observe that, as the number of scenarios increases, the number of iterations decreases for $2BC^*$. This is due to the larger number of multi-cuts inserted in each primal iteration. In contrast to this, the number of iterations for EF increases drastically with the number of scenarios, which explains why instances with more than 20 scenarios are not solvable within the time limit.

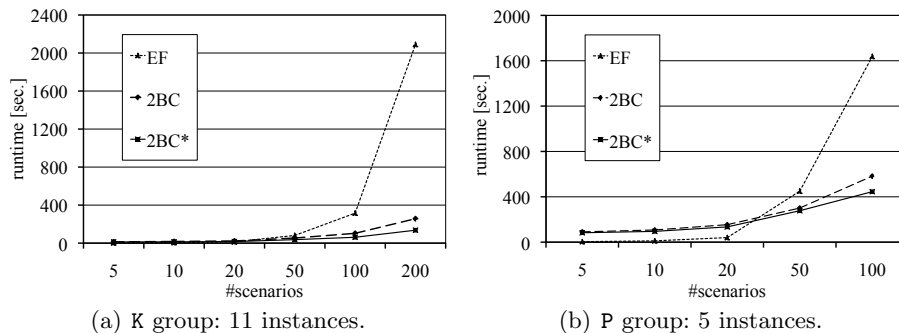


Fig. 1. Average running times in seconds for both two-stage branch-and-cut algorithms $2BC$ and $2BC^*$, and for the extensive formulation of the deterministic equivalent EF .

5 Extensions and Future Work

Gupta et al. [5] also consider the SSTP in which the first stage solution is a tree. Using our above ideas and bi-directing G already for the first stage, we can deduce an even stronger fully directed model that ensures that the first-stage solution is a rooted Steiner arborescence as well. It will be interesting to evaluate the potentially arising benefits.

Along the lines of the *algorithm engineering cycle*, our above approach leaves multiple areas for further improvements: The integration of stronger primal heuristics may lead to further significant speed-ups. A broader set of specifically designed benchmark instances may allow a better insight in the dependencies between input properties and solvability; e.g., it seems to be hard to generate SSTP instances that require integer L-shaped cuts in practice. It is also an open question how to integrate further known strong arborescence constraint classes like flow-balance constraints, as they are not directly valid in our SSTP setting.

References

1. J.R. Birge and F. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34:384–392, 1988.
2. J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1997.
3. S. Chopra and M. R. Rao. The Steiner tree problem I: Formulations, compositions and extension of facets. *Mathematical Programming*, 64:209–229, 1994.
4. A. Gupta, M. Hajiaghayi, and A. Kumar. Stochastic steiner tree with non-uniform inflation. In *APPROX and RANDOM 2007*, volume 4627 of *LNCS*, pages 134–148, 2007.

Instance	K	R _{avg}	OPT*	EF				2BC*			
				t[s]	b&b	gap	#iter	t[s]	b&b	gap	#iter
lin01.53.80	5	4.6	797.0	0.2	1	—	34	2.2	1	—	61
lin01.53.80	10	4.2	633.2	0.7	3	—	59	2.5	3	—	50
lin01.53.80	20	4.6	753.9	5.7	3	—	63	6.9	3	—	52
lin01.53.80	50	4.7	768.9	33.4	3	—	70	10.4	3	—	36
lin02.55.82	5	4.6	476.2	0.1	1	—	24	1.1	1	—	45
lin02.55.82	10	5.3	739.1	1.0	1	—	33	3.0	1	—	47
lin02.55.82	20	5.3	752.2	4.9	1	—	69	4.3	1	—	37
lin02.55.82	50	5.1	732.6	31.2	1	—	70	10.7	1	—	35
lin03.57.84	5	4.4	653.0	0.5	1	—	80	1.9	1	—	55
lin03.57.84	10	5.2	834.7	3.8	7	—	90	8.7	7	—	91
lin03.57.84	20	5.8	854.9	10.8	1	—	92	7.3	1	—	41
lin03.57.84	50	5.5	895.7	103.1	3	—	106	21.3	3	—	43
lin04.157.266	5	10.4	1922.1	140.4	3	—	315	959.2	47	—	567
lin04.157.266	10	9.8	1959.1	415.8	7	—	244	989.2	7	—	339
lin04.157.266	20	9.3	1954.9	5498.7	11	—	833	3016.7	13	—	575
lin04.157.266	50	9.8	2097.7	(2h)	1	19.5	185	5330.2	11	—	269
lin05.160.269	5	10.2	2215.5	282.0	53	—	722	2681.2	35	—	1558
lin05.160.269	10	11.4	2210.2	1866.7	5	—	1130	4096.0	35	—	1502
lin05.160.269	20	11.1	2412.2	(2h)	11	5.6	1060	(2h)	17	4.7	890
lin05.160.269	50	11.6	2297.0	(2h)	1	21.3	210	3627.4	1	—	159
lin06.165.274	5	11.0	1975.8	212.8	53	—	797	760.9	19	—	834
lin06.165.274	10	10.6	1918.7	501.7	5	—	260	808.4	3	—	306
lin06.165.274	20	14.0	2457.6	(2h)	11	—	1099	3222.9	11	—	459
lin06.165.274	50	12.6	2186.8	(2h)	1	22.5	221	2795.5	11	—	215

Table 1. Results for lin instances: within the time limit of two hours, EF was not able to solve most of the instances with 50 scenarios.

5. A. Gupta, R. Ravi, and A. Sinha. LP rounding approximation algorithms for stochastic network design. *Math. of Operations Research*, 32(2):345–364, 2007.
6. D. S. Johnson, M. Minkoff, and S. Phillips. The prize-collecting Steiner tree problem: Theory and practice. In *Proc. 11th ACM-SIAM SODA*, pages 760–769. SIAM, 2000.
7. G. Laporte and F.V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.*, 13:133–142, 1993.
8. I. Ljubić, R. Weiskircher, U. Pferschy, G. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Mathematical Programming*, 105(2-3):427–449, 2006.
9. PCSTP Benchmark: homepage.univie.ac.at/ivana.ljubic/research/pcstp/.
10. Steinlib: steinlib.zib.de/steinlib.php.

Appendix

Proof of Lemma 1.

Proof. We only show that in any optimal solution of (STP_{sd}) , values of X_e variables will not be fractional. So assume that there exists an optimal solution X such that there exists $e \in E$, such that $0 < X_e < 1$. Inequalities (2) imply that for all scenarios $k \in \mathcal{K}$ $z_{ij}^k + z_{ji}^k = 1$. The term in the objective function corresponding to edge e is:

$$c_e X_e + \sum_k p_k q_e^k (1 - X_e) = c_e X_e + q_e^* (1 - X_e) = (c_e - q_e^*) X_e + q_e^*.$$

Since $c_e - q_e^* < 0$, we can obviously reduce the value of the objective function by setting $X_e := 1$, which is a contradiction to X being the optimal solution.

Proof of Lemma 2.

Proof. It is not difficult to see that the \subseteq -relationship holds. To show the strict inequality, consider the following example.

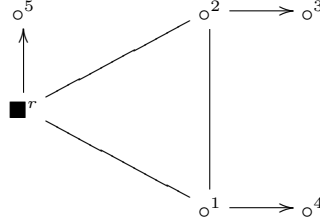


Fig. 2. A network used in the example below. All edge costs are equal to one.

For the network given in Figure 2, we assume that scenarios are assigned a constant inflation factor, σ_k , for all $k \in \mathcal{K}$, so that $q_e^k = \sigma_k c_e$, for all $e \in E$. The following scenario values are given:

$$\begin{aligned} \text{Scenario 1: } & \sigma_1 = 1.5, \quad p_1 = 1/4, \quad R^1 = \{1, 2, 3\}, \\ \text{Scenario 2: } & \sigma_2 = 1.5, \quad p_2 = 1/4, \quad R^2 = \{1, 2, 4\}, \\ \text{Scenario 3: } & \sigma_3 = 3, \quad p_3 = 1/2, \quad R^3 = \{5\}. \end{aligned}$$

The optimal LP-solution of $(SSTP_u)$ sets $X_{r5} = Y_{23}^1 = Y_{14}^2 = 1$ and $Y_{r2}^l = Y_{r1}^l = Y_{12}^l = 1/2$, for $l = 1, 2$. The other variables are set to zero. Therefore, $v_{LP}((SSTP_u)) = 2 \frac{7}{8}$.

On the other hand, this solution is not feasible for the model (STP_{sd}) , which proves the strict inequality in Lemma 2.